

BUNDESREPUBLIK DEUTSCHLAND

CERTIFIED COPY OF
PRIORITY DOCUMENT



Prioritätsbescheinigung über die Einreichung einer Patentanmeldung

Aktenzeichen: 101 28 726.7

Anmeldetag: 13. Juni 2001

Anmelder/Inhaber: Siemens Aktiengesellschaft, 80333 München/DE

Bezeichnung: Programmierschnittstellen zu einer Vermittlungsstelle

IPC: H 04 M 3/00

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprünglichen Unterlagen dieser Patentanmeldung.

München, den 8. September 2005
Deutsches Patent- und Markenamt

Der Präsident
im Auftrag

Agurks

Beschreibung

Programmierschnittstellen zu einer Vermittlungsstelle

5

1. Welches technische Problem wird durch die Erfindung gelöst ?

10 Vermittlungsstellen in Telefonnetzen sind typischerweise geschlossene Systeme, d.h. sowohl die Realisierung der unterstützten Funktionen, wie z.B. Aufbau von Rufen, Vergebührung, Statistik, als auch die Hardwarebasis sind herstellerspezifisch. Netzbetreiber von Telekommunikationsnetzen müssen daher ihre spezifischen
15 Anforderungen an Vermittlungsstellenfunktionen mit den Herstellern absprechen, die diese dann im Auftrag der Netzbetreiber realisieren.

In zunehmendem Maße fordern daher die Netzbetreiber, daß die Telekommunikationsausrüster sowohl als Basis Ihrer
20 Vermittlungsstellen kommerzielle Plattformen verwenden als auch die Funktionen auf diesen Plattformen für eine Steuerung von "außen" öffnen, d.h. der Netzbetreiber selbst oder von ihm beauftragte Programmierer sollen über geeignete Schnittstellen selbst die Funktionen der Plattformen steuern
25 oder erweitern können.

Einer der Hauptgründe für diese Forderung nach Offenheit, ist der Konkurrenzdruck der Netzbetreiber in liberalisierten Netzen. Die Netzbetreiber versuchen sich daher primär durch neue attraktive Mehrwertdienste voneinander zu
30 differenzieren. Dabei ist es wichtig, Betreiberspezifische neue Mehrwertdienste schnell einführen zu können. Die Netzbetreiber fordern daher offene Schnittstellen, damit sie selbst Mehrwertdienste implementieren können.

2. Wie wurde das genannte Problem bisher gelöst ?

Wie oben erwähnt, waren bisher Vermittlungsstellen geschlossene Systeme, d.h. offene APIs sind nicht vorhanden.

- 5 Neue Dienste der Vermittlungsstellen haben die Telekommunikationsausrüster entwickelt und kundenspezifische Entwicklungen im Auftrag eines Netzbetreibers waren aus wirtschaftlichen Gründen nur in begrenztem Maße möglich. Daneben wird von vielen Vermittlungsstellen eine für
- 10 Dienststeuerung standardisierte SS7- Protokollschnittstelle zu externen IN-Systemen unterstützt (INAP - Intelligent Network Application Protocol). Von vielen IN-Herstellern wird für diese IN-Systeme ein Service Creation Environment angeboten, das eine IN-Dienst-Programmerstellung durch den
- 15 Netzbetreiber selbst oder eine von ihm beauftragte SW-Firma ermöglicht. IN Service Creation Systeme haben sich am Markt nicht durchgesetzt. Als ein wesentlicher Grund dafür wird u.a. die bei der Nutzung dieser Systeme erfahrene Komplexität der Dienstprogrammierung angesehen.
- 20 Seit kurzer Zeit gibt es Bemühungen, zusätzlich zu dem standardisierten INAP - Protokoll und anderen standardisierten Protokollen in öffentlichen Telekommunikationsnetzen auch Programmierschnittstellen (Application Programming Interfaces, APIs) für
- 25 Telekommunikationsanwendungen zu standardisieren. Zunächst in Interessengruppen der Industrie, z.B. PARLAY, JAIN, und nun auch in internationalen Standardisierungsgremien wie 3GPP, ETSI und ITU-T. Diese standardisierten APIs bilden die Funktionalität der unterliegenden Protokolle als
- 30 Programmierschnittstelle ab. Inwieweit sich die Nutzung dieser APIs am Markt durchsetzen wird, ist offen.

3. In welcher Weise löst Ihre Erfindung das angegebene technische Problem ?

Gemäß der Erfindung wird eine klassische Vermittlungsstelle mit einer zusätzlichen Plattform auf Basis kommerzieller Hardware (HW) und Software (SW), z.B. das Standard Operating System UNIX, ergänzt, auf der spezielle Module (Applikationsblöcke) implementiert werden, deren Funktionen extern über offene APIs gesteuert werden können. Diese Plattform erlaubt so dem Netzbetreiber die Entwicklung eigener Telefonmehrwertdienste in einer kommerziellen SW Umgebung auf der Basis von Application Programming Interfaces (API) zu Vermittlungsstellenfunktionen. Vermittlungsstelle und kommerzielle Plattform kommunizieren über eine interne Schnittstelle, die nicht standardisiert ist und deren Funktionalität sich nach den Anforderungen der zu unterstützenden SW-Module auf der kommerziellen Plattform richtet. Auf der kommerziellen Plattform können unterschiedliche Applikationen implementiert werden: Protokollumsetzungs-Applikationen, die die standardisierten APIs anbieten können und andere Applikationen, die nicht-standardisierte APIs anbieten können.

Ein Unterschied und Vorteil des erfindungsgemäßen Verfahrens gegenüber existierenden IN SCEs und anderen Systemen, die nur über standardisierte Protokoll-Schnittstellen zu Vermittlungsstellen verfügen, liegt in der Möglichkeit, über die interne Schnittstelle zwischen kommerzieller Plattform und Vermittlungsstelle grundsätzlich freien Zugang zu allen Vermittlungsstellenfunktionen zu haben und diese für die Implementierung der erfindungsgemäßen Applikationen auf der kommerziellen Plattform nutzen zu können.

Ein zweiter Unterschied des erfindungsgemäßen Verfahrens gegenüber existierenden IN SCEs und anderen Systemen, die nur standardisierte APIs bereitstellen, liegt in der Idee, nicht-standardisierte APIs zu höherwertiger Dienstefunktionalität

zur Verfügung zu stellen, wie z.B. den automatischen Aufbau einer Verbindung zwischen zwei PSTN-Teilnehmern, den automatischen Aufbau einer Telefonkonferenz zwischen mehreren Teilnehmern, das Buchen einer automatischen Telefonkonferenz für einen bestimmten Zeitpunkt.

Jeder dieser Dienstapplikationsblöcke hat über die interne Schnittstelle Zugriff zu den Call Processing Funktionen der Vermittlungsstelle. Die Grundfunktion eines Dienstapplikationsblocks ist fest definiert, wie z.B. Aufbau einer Verbindung im PSTN zwischen zwei Teilnehmern die über eine E.164 Adresse erreicht werden können inklusive dem Anlegen einer Begrüßungsansage oder Initialisierung von Gebührentickets für jeden Teilnehmer in der Vermittlungsstelle.

Je Dienstapplikationsblock ermöglichen offene APIs das Aktivieren der jeweiligen Dienstfunktionen sowie die Steuerung einzelner Leistungsmerkmale. Beispielsweise könnte für den Dienstapplikationsblock "Automatischer Aufbau einer Verbindung zwischen zwei PSTN-Teilnehmern" explizit steuerbar sein, welcher der beiden Verbindungsteilnehmer die Gebühren übernimmt, ob zur Begrüßung eine Standardansage oder eine persönliche Ansage gespielt werden soll (erfordert Übertragung der Adresse und Identifikation der persönlichen Ansage) oder ob über das API Statusinformationen der Verbindung zurückgemeldet werden sollen (z.B. Teilnehmer besetzt)

Das erfindungsgemäßen Verfahren beinhaltet also den Ansatz, die Call Processing Funktionen einer Vermittlungsstelle nicht direkt über APIs nach außen zu öffnen, sondern zunächst Dienstapplikationsblöcke auf der zur Vermittlungsstelle gehörenden kommerziellen Plattform zu definieren, in denen das dienstspezifische Interworking mit den existierenden Netzfunktionen gelöst wird und deren Funktionen über offene APIs gesteuert werden können. Netzbetreiber können diese Dienstapplikationsblöcke beliebig aktivieren und kombinieren und so neue Mehrwertdienste kreieren. Dabei kann die eigentliche Dienstlogik des vom Netzbetreiber erstellten

Mehrwertdienstes auf einem beliebigen Server im Netz liegen. Die Dienstapplikationsblöcke mit ihren definierten APIs dienen dem Netzbetreiber dabei als Bausteine zur Implementierung eigener Mehrwertdienste (siehe

5 Ausführungsbeispiel unter 5.)

Das Interworking der Dienstapplikationsblöcke untereinander wird durch einen sogenannten Session Manager Applikationsblock unterstützt. Alle Applikationsblöcke, die über ihr API aktive Dienstanfragen bearbeiten, nehmen eine
10 Registrierung für jeden aktiven Dienstanwender im Session Manager vor. Der Session Manager bietet dafür ein internes API an. Wird eine Dienstanfrage beendet, erfolgt eine Deregistrierung. Der Session Manager hat somit jeweils ein aktuelles Abbild aller aktiven Dienstanfragen. Seine Daten
15 können von jeder Applikation gelesen werden, und unterstützen so das Interworking zwischen verschiedenen Dienstapplikationsblöcken.

Ein wesentlicher Vorteil dieses Verfahrens ist, daß der Anwender der APIs zu Dienstapplikationsblöcken keine
20 Detailkenntnisse über existierende Netzfunktionen haben muß. Insbesondere sind auch alle Details des jeweiligen Signalisierungssystems des Basisnetzes auf der Ebene des API Nutzers vollständig transparent und das interne Zusammenspiel zwischen Vermittlungsstelle und Applikationsblock deckt alle
25 möglichen Interworkingfälle ab. Dieses Verfahren erlaubt die Definition von robusten und leicht handhabbaren APIs, da der festgelegte Umfang jedes Dienstapplikationsblocks bzw. APIs mögliche Fehlerfälle einschränkt und vom Hersteller ausgetestet werden kann.

30 Im Gegensatz zu den erfindungsgemäßen APIs zu höherwertigen Dienstapplikationsblöcken basieren die standardisierten APIs (PARLAY, JAIN, 3GPP, u.a.) auf dem Ansatz, Protokollfunktionen abzubilden. Diese APIs sind sehr komplex und erfordern die Übertragung bzw. Steuerung von sehr vielen
35 Steuerinformationen. Die Realisierung von Mehrwertdiensten auf Basis dieser APIs erfordert Telekommunikationsnetz-Expertenwissen.

Durch die Ergänzung einer klassischen Vermittlungsstellenarchitektur mit einer kommerziellen Plattform, über die APIs zu den Vermittlungsstellenfunktionen zur Verfügung gestellt werden, ist es leichter möglich, die
5 APIs auf Basis der state-of-the Art objektorientierten SW-Technologien wie CORBA, JAVA RMI oder DCOM zu realisieren. Bisherige Vermittlungsstellen basieren oft auf älteren SW-Technologien (z.B. CHILL, ASSEMBLER) die eine Öffnung über
10 objektorientierte basierte APIs erschweren.

Durch die Konzeption von nicht-standardisierten, offenen APIs zu höherwertigen Dienstapplikationsblöcken auf dieser kommerziellen Plattform wird der Kreis der potentiellen
15 Anwender von Telekommunikations-APIs erweitert auf Personen ohne Detailkenntnisse über Telekommunikationsnetzfunktionen.

Im folgenden wird die Erfindung nochmals anhand der Zeichnung erläutert, die eine Figur umfaßt.

20 Die Figur zeigt die prinzipielle Funktionsaufteilung zwischen der Vermittlungsstelle mit integrierter kommerzieller Plattform, die offene APIs anbietet (Backend Server) und und den externen Applikationsservern, die diese APIs nutzen
25 (Frontend Server).

Die kommerzielle Plattform enthält mehrere Applikationsblöcke mit fest definiertem Funktionsumfang, die mittels offenen APIs gesteuert werden können. Die Funktionen der
30 Applikationsblöcke verwenden über eine interne Schnittstelle die Core Call Control Funktionen der unterliegenden Vermittlungsstelle.

Die Mehrwertdienste des Netzbetreibers werden auf separaten Applikationsservern im Netz realisiert. Dabei verwenden sie
35 die offenen APIs der kommerziellen Plattform um Aktionen im Basisnetz zu initiieren (z.B. Aufbau einer Verbindung zwischen zwei Teilnehmern) oder über Ereignisse im Basisnetz

informiert zu werden (z.B. Teilnehmer besetzt, ankommender Ruf). Die offenen APIs werden auf Basis von CORBA genutzt - damit ist die Realisierung der SW auf den Applikationsservern als auch dessen HW unabhängig von SW und HW der kommerziellen Plattform.

Abkürzungen/Referenzen:

| | | |
|----|--------|--|
| | API | Application Programming Interface |
| | CORBA | Common Object Request Broker Architecture |
| 10 | DCOM | Distributed Component Object Model |
| | IN | Intelligent Networks |
| | INAP | Intelligent Network Application Protocol (standardisiert bei ETSI und ITU-T) |
| 15 | JAIN | Industriekonsortium (von der Firma SUN geleitet) : www.java.sun.com/products/jain/ |
| | PARLAY | Industriekonsortium: www.parlay.org |
| | PSTN | Public Switching Telecommunication Network |
| | RMI | Remote Message Invocation |
| | SCE | Service Creation Environment |

Patentansprüche

1. Vermittlungsstelle, mit
einer zusätzlichen Plattform, die auf kommerzieller HW und SW
5 basiert und ein oder mehrere SW-Module enthält, die externen
Rechnern über offene Schnittstellen (API) Applikationen mit
bestimmten Funktionen zur Verfügung stellen, wobei die
genannten SW-Module zur Durchführung der genannten Funktionen
interne Schnittstellen zu den Call-Control-Funktionen der
10 Vermittlungsstelle benutzen.
2. Vermittlungsstelle nach Anspruch 1,
dadurch gekennzeichnet, daß
die genannten offenen Schnittstellen auf der Basis einer
15 objektorientierten SW-Technologie realisiert sind.
3. Vermittlungsstelle nach Anspruch 1 oder 2,
dadurch gekennzeichnet, daß
die genannten offenen Schnittstellen auch lokal auf der
20 kommerziellen Plattform von Applikationen verwendet werden
können um Funktionen anderer Applikationen zu nutzen.
4. Vermittlungsstelle nach einem der Ansprüche 1 bis 3,
dadurch gekennzeichnet, daß
25 ein dediziertes SW-Modul auf der kommerziellen Plattform das
Interworking der genannten Applikationen unterstützt. Das
Interworking SW-Modul stellt dafür eine Plattform-interne
Schnittstelle zur Verfügung.
- 30 5. Vermittlungsstelle nach einem der Ansprüche 1 bis 4,
dadurch gekennzeichnet, daß
es sich bei den genannten Applikationen um
Protokollumsetzungs-Applikationen handelt.

6. Vermittlungsstelle nach einem der Ansprüche 1 bis 4,
dadurch gekennzeichnet, daß
es sich bei den genannten Applikationen um Applikationen für
eine höherwertige Dienstfunktionalität handelt.

Zusammenfassung

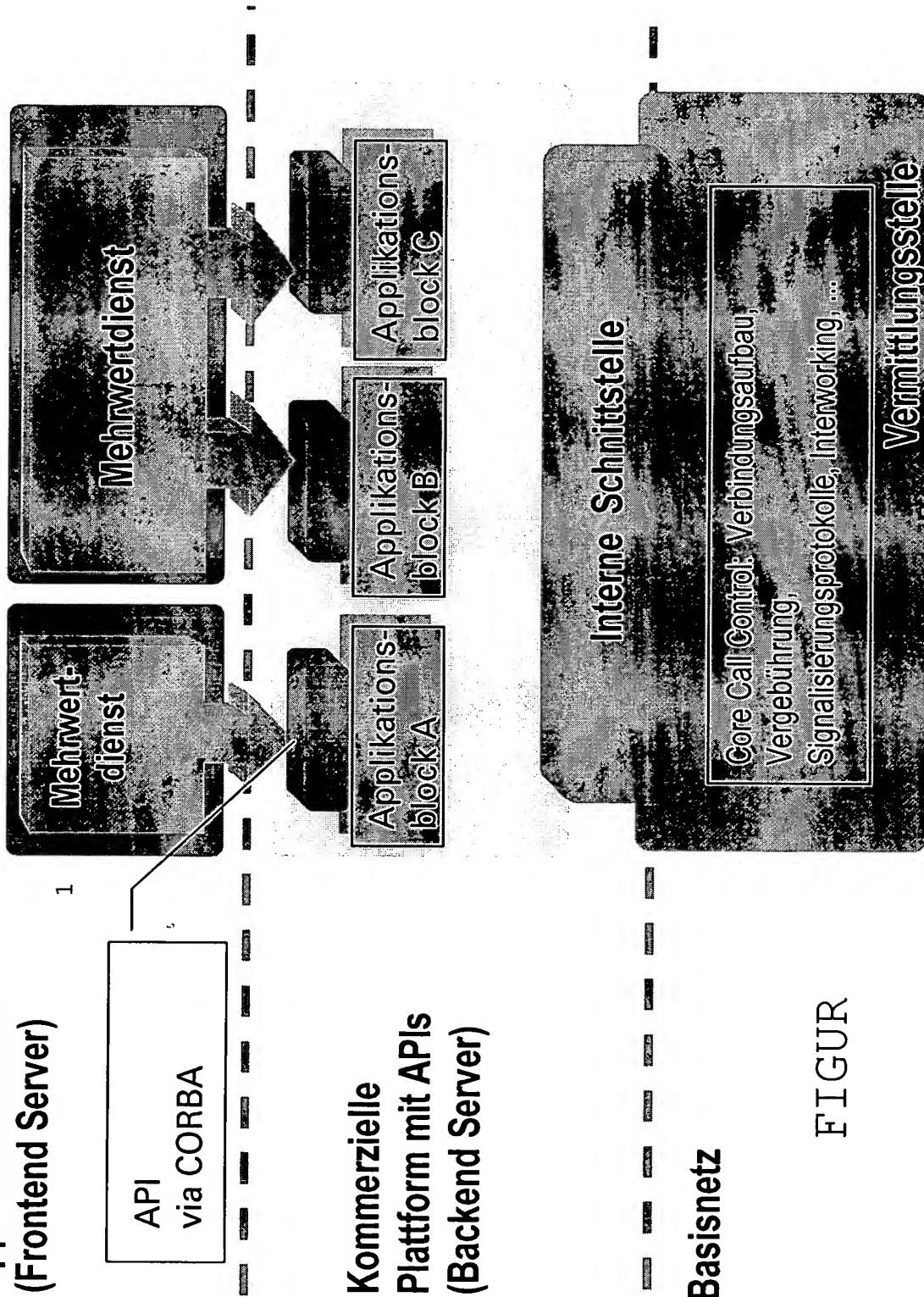
Programmierschnittstellen zu einer Vermittlungsstelle

5 Gemäß der Erfindung wird eine klassische Vermittlungsstelle
mit einer zusätzlichen Plattform auf Basis kommerzieller
Hardware und Software ergänzt, auf der spezielle Module
implementiert werden, deren Funktionen extern über offene
Application Programming Interfaces (API) gesteuert werden
10 können. Diese Plattform erlaubt so dem Netzbetreiber die
Entwicklung eigener Telefonmehrwertdienste in einer
kommerziellen Software Umgebung auf der Basis von APIs zu
Vermittlungsstellenfunktionen.

15 Figur 1

20

Applikationsserver (Frontend Server)



FIGUR